Deterministic Requests: Cacheable OSCORE

`draft-amsuess-core-cachable-oscore`

Christian Amsüss, Marco Tiloca

2021-03-08, IETF 110

Why caching?

- Reduce traffic

  Firmware updates

- Hide traffic

  Firmware updates, again

- Increase reliability
- Decrease latency
- Makes `multicast-notifications` work[1]

  Make protected case as simple as unprotected case

[1] It works without Deterministic Requests. It works *better* with.

Why is this hard in (even Group) OSCORE

$$\left.\begin{array}{l}\text{POST / 2.01}\\ \text{Different PIVs}\end{array}\right\}\quad\text{uncacheable}$$

$$\left.\begin{array}{l}\text{Original KID / PIV unknown}\\ \text{Foreign KID request is untrusted[2]}\end{array}\right\}\quad\text{unverifiable}$$

[2] It'd be a pity if someone requested `/whom-i-know`, but handed you a different request for `/whom-to-trust`

Proposed mechanism[4]

- Dedicated group member: Deterministic Client
- Request-Hash option: Hash of DC sender key || external AAD || plaintext
- "Pairwise" sender key of DC derived from DC sender key and Request-Hash[3]
- Server recognizes DC as requester, builds recipient key from Request-Hash, verifies Request-Hash
- Response bound to request using external AAD (%)

[3] Details pending processing of received comments

[4] including sneak peek at -02

Request-response binding: Details

~~Overriding the Request-KID-Context~~

- Request-Hash as an option in the response
- Request-Hash is Class I for responses
- Request-Hash may be elided from response on the wire transmission but is reconstructed by recipient before OSCORE processing

Limitations

- Only safe requests (GET, FETCH)
- Only resources every group member may access this way
- Algorithms limited to those doing AEAD deterministically

  Currently, all are.
- Security properties traded for cacheability
  - No order between request and response
  - Limited request confidentiality
  - No source authentication
  - No replay protection

Status

- Two implementations interop'd at version between -01 and -02

  The things you learn...
- Addressing pending comments on security.

  Then: review with security in mind.
- Practical testing
- Further WG input?